

POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e
Bioingegneria

Corso di Laurea Magistrale in
Ingegneria Informatica



Bluetooth Low Energy e Risparmio Energetico

Relatore:

PROF. RAFFAELA MIRANDOLA

Correlatore:

DIEGO PEREZ-PALACIN

Tesi di Laurea Magistrale di:

LORENZO PAGLIARI

Matricola n. 798273

Anno Accademico 2014-2015

DEDICHE

dedico il tutto al domo
campione indiscusso....
ovviamente xD

RINGRAZIAMENTI

Ringrazierò qualcuno ahahah....

INDICE

Sommario	ix
1 INTRODUZIONE	1
1.1 The structure of a scientific text	1
1.1.1 Archiving electronic documents: PDF/A	1
2 STATO DELL'ARTE	3
2.1 Risparmio energetico	3
2.1.1 Introduzione	3
2.1.2 Sistemi Self-Adaptive	3
2.2 Bluetooth 4.0 Low Energy	5
2.2.1 Introduzione	5
2.2.2 Caratteristiche generali	8
2.3 Reti Peer-to-Peer	11
2.3.1 Introduzione	11
2.3.2 Modelli di Rete	12
3 PRACTICAL GUIDE TO «CLASSICThesis AT DEIB»	13
3.1 Learn L ^A T _E X	13
3.2 Install L ^A T _E X	13
3.3 Online editor	15
3.4 Building blocks	15
3.4.1 File structure	15
3.4.2 Environments	17
3.5 Contributing to this template	20
BIBLIOGRAFIA	21
i APPENDIX	23
A APPENDIX EXAMPLE	25
A.1 The listings package to include source code	25

ELENCO DELLE FIGURE

Figura 2.1	Schema di compatibilità.	7
Figura 2.2	Compatibilità tecnologie	7
Figura 2.3	Schema di compatibilità.	10
Figura 2.4	Edge dependency	12
Figura 3.1	Thing taken from our master thesis	19

ELENCO DELLE TABELLE

Tabella 2.1	Bluetooth Classic	8
Tabella 2.2	Bluetooth Low Energy	9
Tabella 3.1	Parameters needed for things	19

LISTINGS

Listing A.1	Code snippet with the recursive function to evaluate the pdf of the sum Z_N of N random variables equal to X .	26
-------------	--	----

ACRONIMI

OS operating system

XML eXtensible Markup Language

SOMMARIO

Questo lavoro si posiziona nell'ambito del risparmio energetico e nell'ambito delle applicazioni mobile. Abbiamo studiato una possibile soluzione in grado di trasmettere informazioni attraverso dispositivi mobili, quando si verificano scenari nei quali le comuni reti di telecomunicazione non sono disponibili. Scenari di questo tipo possono essere causati facilmente da forti e avverse condizioni meteorologiche. Grossi nubifragi, forti nevicate o altri eventi di questo genere, possono facilmente danneggiare la struttura della rete di telecomunicazioni, causando l'inattività delle stesse. La nostra idea è stata di progettare un sistema in grado di sfruttare i comuni dispositivi elettronici più diffusi tra le persone e, tramite tecnologie equipaggiate su di essi, per trovare un modo alternativo che non si basasse sulle reti di comunicazione standard, per diffondere informazioni tra le persone. Lavorando con dispositivi mobili, è stato necessario lavorare anche sul discorso energia e consumi. Uno degli obiettivi del sistema che presenteremo in questo elaborato, è quello di regolare il carico di lavoro che si vuole assegnare al dispositivo, in modo da trovare sempre un compromesso tra prestazioni e autonomia residua della batteria. In questo elaborato presenteremo la soluzione da noi proposta, illustrando i principi degli algoritmi utilizzati per guidare la diffusione dell'informazione, i modelli usati per descrivere la rete di dispositivi mobili e la tecnologia di trasmissione utilizzata.

INTRODUZIONE

1.1 THE STRUCTURE OF A SCIENTIFIC TEXT

ciao come va ??

primo ref [14]

1.1.1 *Archiving electronic documents: PDF/A*

PDF/A is an ISO-standardized version of the Portable Document Format (PDF) specialized for the digital preservation of electronic documents. PDF/A differs from PDF by prohibiting features ill-suited to long-term archiving, such as font linking (as opposed to font embedding). The ISO requirements for PDF/A file viewers include color management guidelines, support for embedded fonts, and a user interface for reading embedded annotations.

Universities usually requires this standard but they're also not aware that common programs like MS Word, OpenOffice and so on aren't really able to produce compliant PDFs. In LaTeX, there's some development going on but at the time of writing, the available commands are still too obscure and buggy. So in the end, forget the PDF/A for now.¹

¹ Or DIY and then make a pull request on github :D.

STATO DELL'ARTE

2.1 RISPARMIO ENERGETICO

2.1.1 *Introduzione*

Nel settore dell'IT è ormai da parecchi anni che si guarda al problema dei consumi energetici e alle possibili soluzioni per ridurli. La tendenza verso cui questo settore sta andando è quella di spostare la parte di computazione e di storage sul Web, a carico a grandi aziende che offrono servizi di hosting o clouding. In generale ogni azienda che possiede almeno una server farm medio - grande, si dedica allo studio del risparmio energetico, in quando la spesa per l'energia per il ramo IT non è indifferente ai fini del budget aziendale. Sono stati fatti studi con lo scopo di trovare possibili soluzioni in grado di ridurre gli sprechi. Anche se non ce ne rendiamo conto, nei sistemi IT vi sono molti sprechi di energia. In [14] è illustrato che una stima del costo energetico di un Data Center di Google sia nell'ordine dei milioni di dollari per anno e che il consumo energetico del settore IT a livello mondiale sia stato stimato attorno ai 40 miliardi di dollari nel 2009. Questo fa capire come con l'evolvere della tecnologia, essa diventi sempre più potente e performante, ma necessiti anche di una quantità maggiore di energia. Per questo motivo i discorsi di consumo energetico, della riduzione degli sprechi e dell'ottimizzazione nell'uso delle risorse sono divenuti sempre più importanti e oggetto di ricerca.

2.1.2 *Sistemi Self-Adaptive*

In letteratura si trovano molti studi riguardanti possibili soluzioni per una migliore gestione delle risorse e minimizzazione dello spreco di energia. Questi tipi di sistemi si chiamano Self-Adaptive, poiché riescono ad auto regolare parametri interni con l'obiettivo di consumare il minimo quantitativo di energia necessario. In applicazioni reali, bisogna però tenere conto non solo dei vincoli energetici, ma anche della qualità del servizio. E' logico pensare che un'azienda con un data center offra servizi sul web, quindi queste soluzioni self-adaptive devono tro-

vare un compromesso tra consumo di energia e qualità del servizio (QoS) da garantire. Questo trade-off è il motivo per cui, al giorno d'oggi, non si sono trovate soluzioni definitive. In letteratura vi sono molti articoli riguardanti sistemi auto adattativi con vincoli di QoS, come il [8], [13], [9] e [10]. L'idea di base è di poter gestire la quantità di risorse per offrire un determinato servizio, in modo da allocarne il minor numero possibile garantendo allo stesso tempo una qualità del servizio accettabile senza che l'utente ne percepisca un degrado. Questo tipo di adattamento è ovviamente molto dinamico e molto dipendente dalla distribuzione delle richieste che vengono fatte per quel particolare servizio. Ad esempio, in [10] è discussa la situazione di un server mail aziendale ed è presentato su grafico l'utilizzo del server mail nel tempo, nell'arco di una settimana. Dal grafico risulta che tra un giorno e il successivo, quindi di notte, vi è un uso limitato, mentre nella parte diurna dei giorni lavorativi si hanno alti livelli di utilizzo. Nel week-end invece un uso trascurabile. In questo caso tramite un monitoraggio, si può creare un sistema self-adaptive in grado di reagire in anticipo alle variazioni delle richieste in arrivo al server mail, allocando o deallocando risorse preventivamente. Più l'attività che si cerca di studiare non ha una distribuzione di richieste ben definita o è soggetta ad alta variabilità e magari a imprevedibili momenti di burst di richieste, più sarà difficile creare un sistema auto adattativo in grado di reagire preventivamente nel modo corretto. In questo caso lo studio del trade-off è più complicato e bisogna affidarsi a più euristiche. In generale, il concetto è di avere un sistema in grado di allocare solo le risorse necessarie a gestire l'attuale carico di lavoro col minor spreco di energia e garantendo una certa QoS. Il sistema deve anche poter anticipare un cambiamento nella frequenza di richieste in ingresso al sistema, allocando o deallocando risorse, sempre con l'obiettivo di avere attive solo le risorse necessarie a gestire il carico di lavoro.

Altri esempi di più recente attualità riguardano il mondo mobile. Dall'avvento dei primi smartphone, gli utenti si sono resi subito conto che uno dei problemi di questi dispositivi è l'eccessivo consumo energetico e la conseguente poca autonomia rispetto ai cellulari della precedente generazione. Infatti, le case produttrici implementano su tutti i loro dispositivi mobili moduli di risparmio energia, in grado di agire sul sistema in base all'impostazione scelta. Anche se le impostazioni variano da brand a brand, in generale si può scegliere tra una modali-

tà normale, senza risparmio energetico, una modalità di semi risparmio energetico e una o due modalità di super risparmio energetico in cui viene disabilitato praticamente ogni tipo di servizio non essenziale, lasciando al dispositivo le sue funzioni core e niente altro. Per quanto riguarda i dispositivi mobili ci si sta muovendo verso il Mobile Cloud Computing, come discusso in [11]. Data la potenza di elaborazione con cui sono equipaggiati i dispositivi mobili oggi giorno, essi sono a tutti gli effetti dei computer in miniatura e quindi possono fare tutto; a prova del sempre crescente numero di applicazioni create e immesse sul mercato. Tutto ciò si traduce in un consumo energetico per sostenere l'esecuzione di tutte queste applicazioni e in un problema d'immagazzinamento dati. Tramite il Mobile Cloud Computing, i dispositivi mobili posso scaricare il peso dell'elaborazione di certi dati e la memorizzazione dei file direttamente nel Cloud, con il risultato di risparmiare fino al 25% di batteria e molto spazio di memorizzazione.

Un altro ambito in cui è molto importante lo studio e l'ottimizzazione del consumo energetico è quello dell'hardware. Ad esempio nei microprocessori dei computer, ma non solo, sono presenti moduli in grado di monitorare lo stato del chip, la frequenza di lavoro e altri parametri. Nello specifico il modulo Dynamic frequency/voltage scaling (DVS) permette di regolare il voltaggio e quindi la frequenza di lavoro del processore in maniera dinamica, senza dover spegnere il componente. In [13] sono presentate le specifiche dell'Advanced Configuration and Power Interface (ACPI) e proposte come open standard per la gestione della configurazione e dell'energia di sistemi singoli o insieme di sistemi.

2.2 BLUETOOTH 4.0 LOW ENERGY

2.2.1 Introduzione

La tecnologia Bluetooth 4.0 Low Energy, nome in codice Seattle, è una tecnologia wireless ed è stata rilasciata a metà 2010, col rilascio delle le specifiche relative allo standard; ora alla versione 4.2 [4]. Dato un sempre maggior impiego dei dispositivi Bluetooth in svariati ambiti come l'IT, moduli di rilevamento e sensori, healthcare [5] e altri ancora, questa tecnologia è stata progettata per risolvere uno dei punti deboli delle versioni precedenti: l'alto consumo energetico richiesto. Sulla tecnologia

Bluetooth Low Energy, l'azienda lancia la gamma di prodotti denominati Bluetooth Smart identificando una serie di mercati in cui è richiesta una tecnologia a basso consumo energetico come le nascenti strutture delle smart home, healthcare, sport & fitness. Questa tecnologia è stata progettata e migliorata per tutti questi dispositivi che utilizzano la batteria "a bottone" infatti, alcuni vantaggi presentati dall'azienda sono:

- Basso consumo energetico.
- Possibile autonomia di mesi o anni per sistemi alimentati da batterie a bottone.
- Ampia compatibilità con molti smartphone, computer e tablet.

Sempre nel 2010 si venne a creare un problema di retro compatibilità dei dispositivi Bluetooth Smart con tutti i dispositivi equipaggiati con tecnologie Bluetooth Classic (il Bluetooth Classic è la vecchia tecnologia dalla versione 1 fino a prima della versione 4). E' risultato che la tecnologia Smart era completamente non compatibile con i dispositivi Bluetooth Classic. Per questo motivo nel 2011 l'azienda cambiò il logo del marchio Bluetooth Smart [3], dividendolo in:

- Bluetooth Smart.
- Bluetooth Smart Ready.

L'azienda creò una versione parallela alla Smart, chiamata Smart Ready che implementa l'architettura del Bluetooth 4.0, quindi in grado di comunicare con il Bluetooth sia Smart sia Smart Ready, ma che è anche in grado di comunicare con il Bluetooth Classic, inserendo quell'elemento di retro compatibilità che mancava. Tutti i dispositivi come smartphone, computer e tablet utilizzano tecnologia Bluetooth Smart Ready, mentre dispositivi progettati per funzionare con batterie a bottone o disponibilità energetiche limitate sono equipaggiate solamente con la tecnologia Bluetooth Smart. In figura 2.1 è rappresentato lo schema di compatibilità, mentre in figura 2.2 è riportata la tabella di compatibilità pubblicata dall'azienda.

Tutti gli smartphone, dal 2010 fino ad oggi sono stati equipaggiati con la tecnologia Low Energy Smart Ready. Si è, infatti, constatato un forte incremento nel mercato, di dispositivi che sfruttano tale tecnologia, come i bracciali con tecnologia fitbit, la serie degli i-Watch e gli smartwatch di Android che rendono

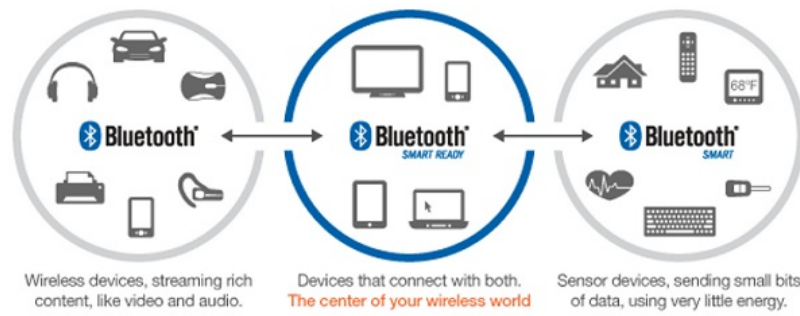


Figura 2.1: Schema di compatibilità tra dispositivi Bluetooth Classic, Smart e Smart Ready.

If your product bears this logo...	It's compatible with products bearing any of these logos...

Figura 2.2: Schema di compatibilità tra tecnologie Bluetooth

l'orologio una vera e propria estensione del telefono. Tutti questi dispositivi sfruttano la tecnologia Bluetooth 4.0 Low Energy rendendo gestibile l'uso di così tante periferiche Bluetooth senza avere drammatiche conseguenze sulla durata delle batterie. Ora l'azienda Bluetooth sta già lavorando a una nuova versione del Bluetooth Low Energy, che è la versione 4.1. Essa mira a ridurre alcuni conflitti di banda con la tecnologia LTE. Infatti, la 4.0 si trova tra le bande 40 e 41 della tecnologia LTE. Per risolvere eventuali conflitti, la nuova versione 4.1 effettua un controllo di banda prima di iniziare le trasmissioni. Uno dei primi smartphone a essere equipaggiato con la nuova versione 4.1 del BLE è stato il Google Nexus 6.

2.2.2 Caratteristiche generali

In questa sezione presentiamo, in via generale, alcune caratteristiche del Bluetooth 4.0 Low Energy, anche con qualche comparazione col predecessore Bluetooth Classic per valutare i miglioramenti e i vantaggi che questa nuova tecnologia offre.

2.2.2.1 Consumo energia

La tecnologia Bluetooth Low Energy presenta un consumo ridotto di energia rispetto alla versione precedente, il Bluetooth Classic. Per il Bluetooth Classic vi sono tre classi distinte in cui sono categorizzati i dispositivi, con le relative distanze raggiunte e i consumi associati, mentre per il Bluetooth Low Energy, l'azienda Bluetooth non ha specificato alcun valore massimo di distanza o consumi e quindi il tutto è lasciato come libera scelta ai produttori dei trasmettitori. Questo si traduce ovviamente in una stima per il calcolo del consumo medio di questa nuova tecnologia. Nella tabella 2.1 sono riportati i valori di consumi energetici e distanze, relativi al Bluetooth Classic [2], mentre nella tabella 2.2 sono riportati i valori per il Bluetooth Low Energy riportati nelle specifiche ufficiali [4] e anche in [1]. Come si può vedere il consumo di energia del Low Energy è ridotto di almeno un ordine di grandezza rispetto al precedente standard Classic.

Come discusso in [7], il consumo molto basso da parte del Low Energy, permette a dispositivi alimentati con batterie a bottone, un ciclo di vita che varia tra i 2 giorni e i 14 anni. Inoltre rispetto allo standard Classic, che consente un massimo di 7 dispositivi slave per ogni master, la tecnologia Low Energy offre più flessibilità rendendo questo valore dipendente dall'applicazione e può variare tra 2 e 5.917 slave per master. Per quanto riguarda la distanza di trasmissione del Bluetooth Low Energy, è stato rilevato essere in media attorno ai 50 metri.

	POTENZA (mW)	POTENZA (dBm)	DISTANZA (m)
Classe 1	100	20	~100
Classe 2	2.5	4	~10
Classe 3	1	0	~1

Tabella 2.1: Caratteristiche Bluetooth Classic.

POTENZA MASSIMA	POTENZA MININIMA	DISTANZA
10 mW (10 dBm)	0.01 mW (-20 dBm)	~50 m

Tabella 2.2: Caratteristiche Bluetooth Low Energy.

2.2.2.2 *Link Layer*

Come descritto nelle specifiche ufficiali [4], l'operatività del Link Layer può essere descritta dai seguenti stati:

- Stato di Standby.
- Stato di Advertising.
- Stato di Scanning.
- Stato di Initiating.
- Stato di Connection.

Questi rappresentano i nodi dell'automa a stati finiti che modella il comportamento del Link Layer. Ogni macchina a stati può essere in un solo stato alla volta, ma un Link Layer può avere più istanze di macchina a stati, se il suo hardware lo consente. È necessario però che almeno una delle sue macchine sia in grado di entrare nello stato di Advertising o nello stato di Scanning. Nel caso di macchine a stati multiple, vi sono restrizioni sulle combinazioni possibili di stati attivi tra tutte le macchine a stati; per ulteriori dettagli si faccia riferimento alla documentazione ufficiale [4].

Lo stato di Standby è uno stato in cui il dispositivo non trasmette e non riceve alcun pacchetto. Questo stato è raggiungibile da qualsiasi altro stato.

Lo stato di Advertising, trasmetterà pacchetti di advertising e, se possibile, ascolterà e risponderà a quei pacchetti trasmessi in risposta al suo pacchetto di advertising. Dispositivi nello stato di Advertising sono chiamati advertiser e lo stato di Advertising è raggiungibile dallo stato di Standby.

Lo stato di Scanning è uno stato di osservazione. Un dispositivo nello stato di Scanning rimane in ascolto per i pacchetti di advertising. Un dispositivo nello stato di Scanning è chiamato scanner e lo stato di Scanning è raggiungibile solo dallo stato di Standby.

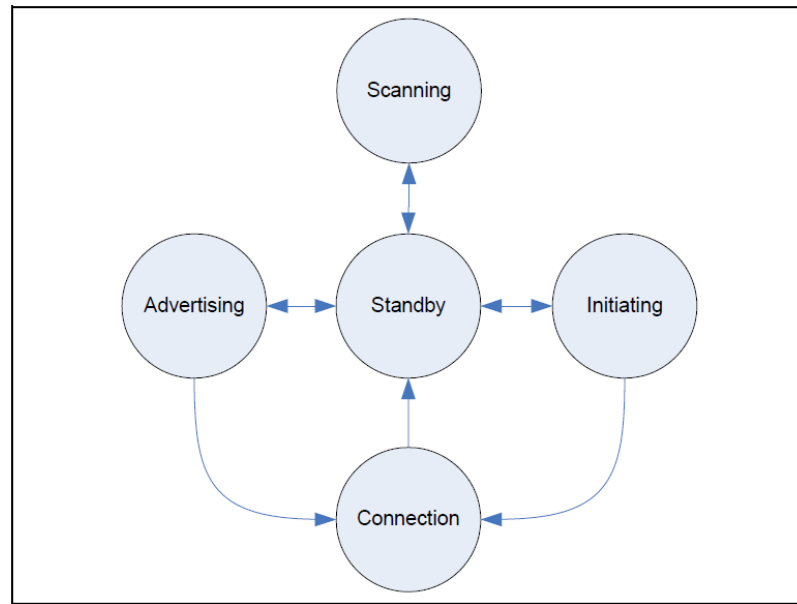


Figura 2.3: Schema di compatibilità tra dispositivi Bluetooth Classic, Smart e Smart Ready.

Un Link Layer nello stato di Initiating rimarrà in ascolto per pacchetti di advertising trasmessi da determinati dispositivi e risponderà a determinati pacchetti se ha l'intenzione di aprire una connessione con quel dispositivo mittente. Un dispositivo in stato di Initiating è chiamato initiator. Lo stato di Initiating è raggiungibile dallo stato di Standby.

Lo stato di Connection può essere raggiunto sia dallo stato di Advertising sia dallo stato di Initiating. Lo stato di Connection si divide a sua volta in due ruoli:

- Ruolo Master.
- Ruolo Slave.

Se raggiunto dallo stato di Advertising si entrerà nello stato di Connection col ruolo di Slave, mentre se si raggiunge lo stato di Connection dallo stato di Initiating si entra nello stato Connection col ruolo di Master. Un singolo dispositivo col ruolo di Slave può comunicare con un solo Master alla volta.

In figura 2.3 è riportato lo schema della macchina a stati del Link Layer.

2.3 RETI PERR-TO-PEER

2.3.1 Introduzione

La nostra ricerca si è orientata su modelli di reti in grado di rappresentare reti Peer-to-Peer, chiamate anche Peer2Peer o P2P. Una rete P2P è una rete in cui non vi è una struttura gerarchica e ogni nodo è considerato allo stesso livello di tutti gli altri. Senza la presenza di nodi più importanti o nodi rappresentati centri di conoscenza della rete, nessun nodo può avere una visione completa della rete stessa e non può sapere se la porzione di rete da lui vista rappresenta l'intera rete. Ogni nodo quindi, ha una visione parziale e locale dell'overlay della rete. Le topologie di rete sono rappresentate da grafi bidirezionali, poiché i canali di comunicazione tra i nodi di una rete P2P per le situazioni analizzate, sono bidirezionali. Il modello di rete P2P è un'architettura logica di una rete di nodi paritari, senza alcuna struttura Client-Server fissi [<https://it.wikipedia.org/wiki/Peer-to-peer> da cambiare con il riferimento al libro !!]; ogni nodo è paritario a tutti gli altri, infatti, ogni nodo è chiamato peer. La struttura Client-Server è creata solo nel momento di dover instaurare una connessione tra due nodi, ma più che Client-Server, sarebbe più corretto definirla come Mittente-Destinatario perché non vi sono compiti o azioni predefinite e pre-allocate nelle due parti. Può essere quindi definita una struttura particolare della struttura generica Client-Server.

La principale applicazione di questo modello di rete è stata ed è tuttora quella della condivisione dei file (file sharing), per la quale sono nati tanti sistemi quali Gnutella, FastTrak, Napster, eMule, la rete Torrent, Freenet etc.

Alcune caratteristiche delle reti Peer-to-Peer:

- *Basso costo d'implementazione*: non è richiesto l'uso di potenti macchine Server, ma è sufficiente che ogni peer possa sostenere le transazioni dell'utente locale e degli altri peer che vogliono connettersi a lui.
- *Amministrazione decentralizzata*: non vi è un server centralizzato di stoccaggio delle informazioni, ma le informazioni sono in possesso dei singoli utenti, localmente, che poi mettono a disposizione della rete.
- *Maggiore velocità di trasmissione*: non avendo un Server centralizzato cui tutti i Client si devono connettere per avere

un'informazione, causando un calo nella velocità di trasferimento, in un'architettura P2P l'informazione può essere reperita anche da più nodi contemporaneamente, infatti, è possibile reperire parti diverse della stessa informazione da nodi diversi e alla fine riassemblare il tutto, col grosso vantaggio di poter avere l'informazione in tempi brevi.

- *Sicurezza*: senza la presenza di Server centralizzati, ogni nodo deve garantire per se e per i contenuti che distribuisce, inoltre è anche esposto a ciò che riceve dalla rete che non è controllato da nessuna terza parte.

2.3.2 Modelli di Rete

Poiché abbiamo preso in considerazione l'uso dei grafi, introduciamo due parametri riguardanti i grafi che ne descrivono alcuni aspetti. Essi sono la *edge dependency* e la *degree variance*.

- **Edge Dependency**: anche chiamata interdipendenza tra archi. Dato un grafo come in figura 2.4, la edge dependency è definita come la probabilità subordinata che esista un arco tra gli stati $S_i \sim S_j$, sapendo che esiste un arco tra gli stati $S_i \sim S_k$ e tra $S_k \sim S_j$. Formalmente, questa probabilità si esprime così $P(S_i \sim S_j \mid S_i \sim S_k, S_k \sim S_j)$ e si può considerare alta se è maggiore della sua probabilità semplice $P(S_i \sim S_j)$.

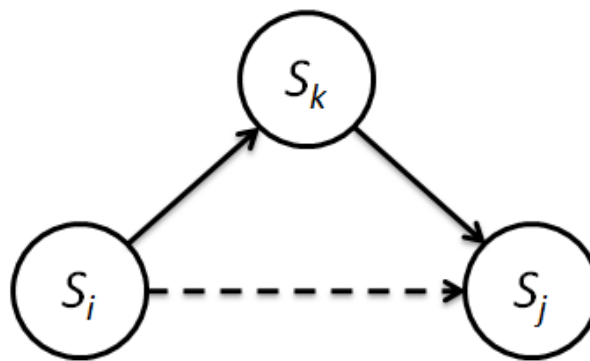


Figura 2.4: Esempio di edge dependency.

PRACTICAL GUIDE TO «CLASSICTHESIS AT DEIB»

This template is ready to be used when writing a thesis at Dipartimento di Elettronica, Informazione e Bioingegneria. It is a modified version of Classic Thesis by André Miede that can be found here <http://code.google.com/p/classicthesis/>.

3.1 LEARN L^AT_EX

L^AT_EX is a document preparation system and document markup language. It is widely used for the communication and publication of scientific documents in many fields, including mathematics, physics, computer science, statistics, economics, and political science.

L^AT_EX users are weird people who care about the ligature between «f» and «i» and gets pissed off every time they look at a MS Word document. Nevertheless, they can explain themselves very well as shown in some beautiful guides for the L^AT_EX world. My preferred one for beginners is «The Not So Short Introduction to L^AT_EX 2_ε», which can be found [here](#).¹ For italians I also strongly suggest «L'arte di scrivere con L^AT_EX», that can be found [here](#).² It contains everything needed, however I suggest the reading of chapter 3 for a short introduction. «ClassicThesis» is another guide of the same author that can be useful, download it [here](#).³

3.2 INSTALL L^AT_EX

If you don't have already a L^AT_EX system installed, this section will explain everything you need. The easiest way to get L^AT_EX is to install TeXLive, which works on all operating systems (OSs). In <https://www.tug.org/texlive/> you find the instructions and the files needed - and also get in touch with minimalism of T_EXusers.

¹ <http://www.ctan.org/pkg/lshort>

² http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf

³ http://www.lorenzopantieri.net/LaTeX_files/ClassicThesis.pdf

Then you will need an editor: I strongly recommend TeXworks because it's very simple and available on all the platforms. Also you don't need to install it, it's already included in TeXLive. The official documentation of TeXworks is available [here](#);⁴ I strongly recommend the reading of chapter 3. Alternatively you can read an italian manual: [profs.sci.univr.it/...](#) (just 13 pages, read it!).⁵

After opening TeXworks, I strongly suggest to set these two additional things:

- open Preferences, then go the Composition tab: in the second box there, the «Process instruments», push the plus button. In the window just opened, write Biber in the «Name» field, biber in the «Program» field (lowercase!) and then press the plus button to add the argument `$basename;`
- again in the same window, set «Hide console output» to «never».

Then just test the installation of the template:

- A. go into the template home folder;
- B. open the file `ClassicThesis_DEIB.tex`;
- C. select pdfLaTeX from the dropdown menu in the top right of the TeXworks window;
- D. press the rounded green button: it compiles the `.tex` file for the first time and open the resulting `.pdf`;
- E. select Biber from the same dropdown menu and press again the green button: this compiles the bibliography, a thing you need to repeat only when you change the file `Bibliography.bib`;
- F. select pdfLaTeX again and recompile: this is needed to build indices and crossreferences;

The above compilation procedure is the standard way to translate the \LaTeX code into pdfs.

⁴ <https://docs.google.com/file/d/0B5iVT8Q7W44pMk1WSFRKcDRlMU0/preview>

⁵ If you already have a preferred editor, just keep using yours.

3.3 ONLINE EDITOR

If the above procedure seems too difficult to you and you have an internet connection always available, you might think to use an online editor. The best choice at the time of writing is <http://sharelatex.com> where you can even find this template after registration to the site by looking for «Classic Thesis At DEIB». Your project will be saved on their server but you can also download them. The platform allows up to two authors for free accounts.

There is no need to provide instructions for its use since the website has them. They also have an online \LaTeX guide which is also very useful.

3.4 BUILDING BLOCKS

3.4.1 *File structure*

The template is organized in multiple file and folders:

- A. `ClassicThesis_DEIB.tex` is the main file to be compiled, found in the root folder. You should just add the source filenames you want to include and any hyphenation you need to explicitly specify.
- B. `classicthesis-config.tex` contains options that can be chosen for this template, like the draft one that prints date and time at the bottom of every page. It contains also the definition for the title, the author and others stuff displayed in the titlepage. Comments within the file should guide you.⁶ Take a look at it!
- C. `Bibliography.bib` is the *Bibtex* database: it is a normal textfile where you should put books and articles read;
- D. `Chapters` contains the files for the main chapters of your thesis; this is where you will add the chapters text, as well these very words in line 41 of the file `Conclusion.tex`;
- E. `CodeFiles` contains any code snippet you want to include in your thesis with the environment `listings`; it might be some relevant Matlab or C code, as well as long bash scripts;

⁶ comments are the rows starting with %.

- F. `FrontBackmatter` contains various files that are included in the main one to produce abstract, titlepages, acknowledgements, Follow the instructions below to modify them in order to suits your needs;
- G. `Images` contains the `.pdf` or `.png` versions of the images of the thesis. A `sources` subfolder is also provided for keeping things well organized.

To modify abstract, preface, acknowledgements and acronyms, you need to go into the folder `FrontBackmatter` where you will find the following:

`ABSTRACT.TEX` contains the text displayed as «abstract» and «sommario» just after the list of figures, tables, etc. Modify the text and leave the rest.

`ACKNOWLEDGMENTS.TEX` contains the text put just before the table of contents. Modify the text to suit your needs.

`ACRONYMS.TEX` contains the environment `acronym` with the definition of all the acronyms that will be used within the text. Add your own to the list and put the longest as parameter of the environment.

`AUTOPARTS` folder contains things that should work without your intervention. Forget them.

`DEDICATION.TEX` same usage and structure as `Acknowledgements.tex`.

`ESTRATTO.TEX` Politecnico di Milano requires an italian long excerpt of theses written in foreign languages.

`FRONTESPIZIO.TEX` and `FrontespizioIT.tex` are the cover page in english and italian, respectively. Politecnico di Milano requires the italian version of the english cover, so there it is. Both should work perfectly if you modify section 2 of the file `classicthesis-config.tex`, but you may not like the style so modify them as you prefer.

`PREFACE.TEX` same usage and structure as `Acknowledgements.tex`.

`PUBLICATION.TEX` same usage and structure as `Acknowledgements.tex`, but not included by default. Activate it by uncommenting the relevant line in `ClassicThesis_DEIB.tex`.

`RETROFRONTESPIZIO.TEX` contains the colophon. In most cases is fine as it already is.

3.4.2 Environments

In addition to common L^AT_EX environments, this thesis is set to use:

- `\begin{aenumerate}` to produce an `\enumerate` with letters instead of numbers, as in the file list above;
- `\blockquote[][\textwidth]{}` to «produce a citation with reference to author and page» [see 6, p. 111]. If the citation is longer than two rows is indented. This is provided by the package `csquotes`, which settings are in `classicthesis-config.tex`. The package also provides `\enquote{the citation}` that produces «correct citation style» according to the language in use.
- `\ac{}` and its variations, defined by package `acronyms`, provide nice handling for acronyms, like eXtensible Markup Language (XML), produced with the code `\ac{XML}`. List them within the environment `acronym` in the file `FrontBackmatter/Acronyms.tex`.
- the so called semi-dynamic referencing for chapter, sections, subsections, appendices, figures, tables and equations. They are a set of commands like `\myChap{label_key}` that produce things like `capitolo ??`. There are also capital versions of the commands (`\MyChap{}` produces `Capitolo ??`). They need a `\label{name}` anchor next to the referred thing.
 - `\myChap` for chapters;
 - `\mySec` for sections;
 - `\mySubsec` for subsections;
 - `\myAppendix` for appendices;
 - `\myFig` for figures;
 - `\myTab` for tables;
 - `\myEq` for equations;
- references to bibliography are produced in the usual way with `\cite{bib_key}` [6] and its variations `\citeauthor{bib_key}`, `\citetitle{bib_key}` and others.
- figures are handled usually with the code

The command `graffito` is used to put some text here, usefull to underline important things before long paragraphs.

```

\begin{figure}
\centering
\includegraphics[width=\columnwidth]{Images/your_image_name.pdf}
\caption[Short description]{Long description.}
\label{fig:a_name}
\end{figure}

```

which produces things like figura 3.1. Of course, you need to put the image file `your_image_name.pdf` in folder `Images/`.

- tables are produced with

```

\begin{table}[tb]
\footnotesize
\centering
\begin{tabularx}{0.8\textwidth}{llrcl}
\toprule
\tableheadline{l}{Algorithm} &
\tableheadline{l}{Parameter} &
\tableheadlineMore{3}{c}{Suggested Values} \\
\midrule
\tablefirstcol{l}{Any}
& \acs{NFE} & & $10\,000$ & & $ & $ \div $ & & $200\,000$ \\
& Population Size & & $10$ & & $ & $ \div $ & & $1000$ \\
\midrule
\tablefirstcol{l}{\ac{GDE3}}
& \ac{DE} step size & & $0.0$ & & $ & $ \div $ & & $1.0$ \\
& Crossover rate & & $0.0$ & & $ & $ \div $ & & $1.0$ \\
\bottomrule
\end{tabularx}
\caption[Short description]{Long description.}
\label{tab:MOEAandParameters}
\end{table}

```

which produces tabella 3.1. `\myfloatalign`, `\tableheadline{}` and its variation `\tableheadlineMore{ }{ }` and `\tablefirstcol{ }` are used to give a common style to all tables in the document. They are defined in `classicthesis-config.tex`.⁷

- equation are produced in classic L^AT_EX way and they turn out be something like this

$$\nabla \mathbf{q}_s = \mathbf{U}(x, y) - \mathbf{b}_t \quad (3.1)$$

⁷ Also do not forget footnotes, created by `\footnote{}`, which should be placed after the punctuation mark.

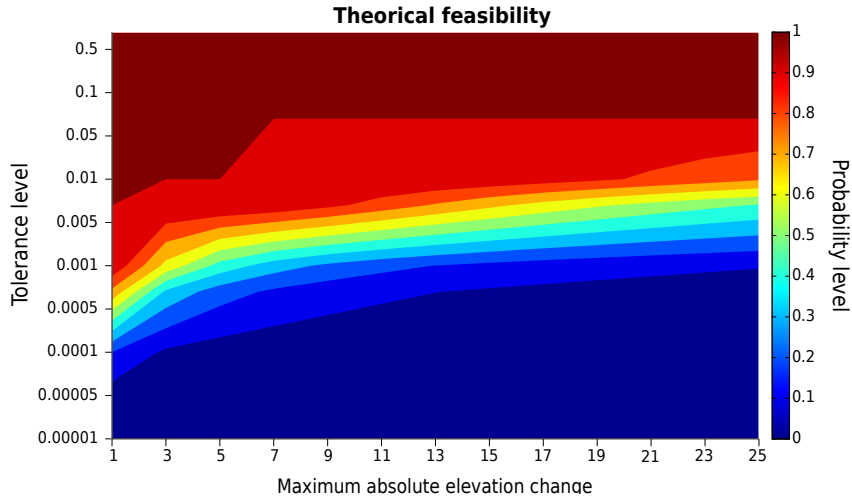


Figura 3.1: Thing taken from our master thesis whose meaning have been completely forgotten.

ALGORITHM	PARAMETER	SUGGESTED VALUES		
Any	NFE	10 000	÷	200 000
	Population Size	10	÷	1000
GDE ₃	DE step size	0.0	÷	1.0
	Crossover rate	0.0	÷	1.0

Tabella 3.1: Parameters needed for things that are not needed anymore themselves.

3.5 CONTRIBUTING TO THIS TEMPLATE

Suggestion and improvements are welcome at <https://github.com/Lordmzn/ClassicThesis-at-DEIB> or via email at emanuele.mason@polimi.it or andrea.cominola@polimi.it.

BIBLIOGRAFIA

- [1] Tiberto Alessia. «Confronto tra Bluetooth Basic Rate e Bluetooth Low Energy». Bachelor Degree. institution. URL: http://tesi.cab.unipd.it/44150/1/tesi_tibertoa.pdf (cit. a p. 8).
- [2] *Bluetooth Basics*. Bluetooth SIG. URL: <http://www.bluetooth.com/Pages/Basics.aspx> (cit. a p. 8).
- [3] *Bluetooth Brand*. Bluetooth SIG. URL: <http://www.bluetooth.com/Pages/Bluetooth-Brand.aspx> (cit. a p. 6).
- [4] *Bluetooth Core Specification 4.0*. URL: <https://www.bluetooth.org/en-us/specification/adopted-specifications> (cit. alle pp. 5, 8, 9).
- [5] *Bluetooth smart marks*. Bluetooth SIG. URL: <https://www.bluetooth.org/en-us/bluetooth-brand/smart-marks> (cit. a p. 5).
- [6] R. Bringhurst. *The Elements of Typographic Style*. Version 3.2. Point Roberts, WA, USA: Hartley & Marks Publishers, 2008 (cit. a p. 17).
- [7] Gomez Carles, Oller Joaquim e Paradells Josep. «Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology». In: *Sensors 2012* 12 (9 ago. 2012), pp. 1587–1611 (cit. a p. 8).
- [8] Perez-Palacin Diego, Mirandola Raffaella e Merseguer José. «Accurate Modeling and Efficient QoS Analysis of Adaptive Systems under Bursty Workload». In: *JOURNAL OF LATEX CLASS FILES* 6.1 (gen. 2007) (cit. a p. 4).
- [9] Perez-Palacin Diego, Mirandola Raffaella e Merseguer José. «On the relationships between QoS and software adaptability at the architectural level». In: *Journal of Systems and Software* 87 (gen. 2013), pp. 1–17 (cit. a p. 4).
- [10] Perez-Palacin Diego, Mirandola Raffaella e Merseguer José. «QoS and energy management with Petri nets: A self-adaptive framework». In: *Journal of Systems and Software* 85 (12 dic. 2012), pp. 2796–2811 (cit. a p. 4).

- [11] Niyato Dusit et al. «A survey of mobile cloud computing: architecture, applications, and approaches». In: *Wireless Communications and Mobile Computing* 13 (18 dic. 2013), pp. 1587–1611 (cit. a p. 5).
- [12] D. E. Knuth. «Computer Programming as an Art». In: *Communications of the ACM* 17.12 (1974), pp. 667–673 (cit. a p. 25).
- [13] Marzolla Moreno e Mirandola Raffaella. «Dynamic power management for QoS-aware applications». In: *Sustainable Computing: Informatics and Systems* 3 (4 feb. 2013), pp. 231–248 (cit. alle pp. 4, 5).
- [14] Ranganathan Parthasarathy. «Recipe for Efficiency: Principles of PowerAware Computing». In: *Communications of the ACM* 53.4 (apr. 2010), pp. 60–67 (cit. alle pp. 1, 3).

Parte I

APPENDIX

APPENDIX EXAMPLE

*We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.*

— Knuth, «Computer Programming as an Art», 1974

A.1 THE `listings` PACKAGE TO INCLUDE SOURCE CODE

Source code is usually not part of the text of a thesis, but if it is an original contribution it makes sense to let the code speak by itself instead of describing it. The package `listings` provides the proper layout tools. Refer to its manual if you need to use it, an example is given in listing [A.1](#).

Listing A.1: Code snippet with the recursive function to evaluate the pdf of the sum Z_N of N random variables equal to X .

```

1 std::vector<int> values_of_x(number_of_values_of_x,
    min_value_of_x);
3 for (unsigned int i = 1; i < number_of_values_of_x; i++) {
    values_of_x[i] = values_of_x[i - 1] + 1;
5 }
    prob_x = 1.0 / number_of_values_of_x;
7 std::vector<std::vector<double> > p_z;
    for (unsigned int idx = 0; idx < p_z.size(); idx++) {
9         p_z[idx] = std::vector<double>(
            (max_value_of_x * (idx + 1) - min_value_of_x
11             * (idx + 1)) + 1, INIT_VALUE);
    }
13
    double prob(int Z, int value_of_z) {
15         if (value_of_z < min_value_of_x * Z ||
            value_of_z > max_value_of_x * Z) {
17             return 0.0;
        }
19         if (value_of_z < min_value_of_z ||
            value_of_z > max_value_of_z) {
21             return 0.0;
        }
23         int idx_value_of_z = -(min_value_of_z - value_of_z);
        int idx_N = Z - 1;
25         if (p_z[idx_N][idx_value_of_z] == -2.0) {
            if (Z > 1) {
27                 double pp = 0.0;
                for (unsigned int i = 0; i < number_of_values_of_x; i
                    ++){
29                     pp += prob(Z - 1, value_of_z - values_of_x[i], p);
                }
31                 p_z[idx_N][idx_value_of_z] = prob_x * pp;
            } else {
33                 if (Z == 1) {
                    for (unsigned int j = 0; j < number_of_values_of_x;
                        j++){
35                         if (value_of_z == values_of_x[j]) {
                            p_z[idx_N][idx_value_of_z] = prob_x;
37                             break;
                        }
                    }
39                 }
            }
41             if (p_z[idx_N][idx_value_of_z] == INIT_VALUE) {
                p_z[idx_N][idx_value_of_z] = 0.0;
43             }
        }
45    }
    return p_z[idx_N][idx_value_of_z];
47 }

```